

“REPOSITORIO DE CÓDIGOS ENTRE PROFESOR Y ALUMNOS”

Barbosa Moreno
Hayde Victoria

**Ingeniería de
Software**

Índice

Índice de ilustraciones.....	4
Índice de tablas.....	5
Índice de ecuaciones.....	5
1. Introducción	6
1.1 Descripción general del problema.....	6
1.2 Objetivos	6
1.3 Estructura de la memoria.....	6
2. Análisis del Sistema	7
2.1 Objeto.....	7
2.2.1 Generalidades.....	7
2.2.2 Perfiles de usuarios.....	8
2.2.3 Objetivos	9
2.2 Funcionalidad	10
2.2.1 Descripción	10
2.3.2 Paradigma.....	10
2.2.2 Recursos.....	11
2.4 Métricas y estimaciones.....	13
2.4.1 Valores de Ajuste de Compatibilidad	13
2.4.2 Puntos de función.....	15
2.4.3 Método (COCOMO)	16
2.4.4 Ecuaciones del modelo COCOMO.....	17
2.5 Costes	19
2.5.1 Costes Hardware	19
2.5.2 Costes Recursos Lógicos.....	19
2.5.3 Costes Recursos Humanos.....	19
2.6 Evaluación de riesgos	20
2.6.1 Registro de riesgos.....	20
2.7 Planeación del proyecto.....	21
2.7.1 Diagrama de Gantt.....	21
3. Análisis de Requisitos	22

3.1	Introducción	22
3.2	Requisitos funcionales	22
3.2.1	RF1: Restricciones tipo de cuenta	22
3.2.2	RF2: Creación y publicación de contenidos	22
3.2.3	RF3: Gestión de usuarios.....	22
3.3	Requisitos no funcionales	22
3.3.1	Requisitos de rendimiento	23
3.3.2	Requisitos de interfaz.....	23
3.4	Modelado de casos de uso	23
3.4.1	Introducción	23
3.4.2	Casos de uso de la aplicación.....	24
4.	Diseño del Proyecto	35
4.1	Introducción	35
4.2	Arquitectura del sitio web	35
4.3	Interfaz grafica	36
4.3.1	Mock-Ups.....	36
5.	Codificación	42
5.1	Introducción	42
5.1.1	XHTML	42
5.1.2	CSS	42
5.1.3	Java Script	42
5.1.4	PHP.....	43
5.1.5	AJAX	43
5.2	Organización final de los archivos	43
5.2.1	Carpeta profesor	44
5.2.2	Carpeta recursos	44
5.2.3	Carpeta usuarios	45
5.3	Librerías externas	46
5.3.1	Syntax Highlighter de Alex Gorbachev.....	46
5.3.2	jQuery	46
6.	Pruebas.....	47

6.1 Introducción	47
6.2 Tipos de pruebas	47
6.3 Pruebas realizadas	48
7. Publicación del Sistema	51
7.1 000WebHost	51
7.2 ingsw2016.netai.net	51
7. Conclusiones.....	52
8.1 Conclusiones finales	52
8.2 Objetivos conseguidos	52
8.3 Problemas encontrados	52
8.4 Posibles ampliaciones	52

Índice de ilustraciones

Ilustración 1 Jerarquía de usuarios.....	8
Ilustración 2 Modelo en Cascada	11
Ilustración 3 Recursos de Hardware	12
Ilustración 4 Recursos lógicos	12
Ilustración 5 Recursos humanos	13
Ilustración 6 Diagrama de Gantt	21
Ilustración 7 Caso de uso No.1	24
Ilustración 8 Caso de uso No.2.....	25
Ilustración 9 Caso de uso No.3.....	26
Ilustración 10 Caso de uso No.4.....	27
Ilustración 11 Caso de uso No.5.....	28
Ilustración 12 Caso de uso No.6.....	29
Ilustración 13 Caso de uso No.7	30
Ilustración 14 Caso de uso No.8.....	31
Ilustración 15 Caso de uso No.9.....	32
Ilustración 16 Caso de uso No.10.....	33
Ilustración 17 Caso de uso No.11	34
Ilustración 18 Arquitectura de datos	35
Ilustración 19 Mock up No. 1	36
Ilustración 20 Mock up No. 2	36
Ilustración 21 Mock up No. 3	37
Ilustración 22 Mock up No. 4	37
Ilustración 23 Mock up No. 5	38
Ilustración 24 Mock up No. 6	38
Ilustración 25 Mock up No. 7	39
Ilustración 26 Mock up No. 8	39
Ilustración 27 Mock up No. 9	40
Ilustración 28 Mock up No. 10	40
Ilustración 29 Mock up No. 11	41
Ilustración 30 Arquitectura de archivos - raíz.....	43
Ilustración 31 Arquitectura de archivos - /profesor.....	44
Ilustración 32 Arquitectura de archivos - /recursos.....	44
Ilustración 33 Arquitectura de archivos - /usuarios	45
Ilustración 34 Vista general online	51

Índice de tablas

Tabla 1 Costes Hardware	19
Tabla 2 Costes recursos lógicos	19
Tabla 3 Costes recursos humanos.....	19
Tabla 4 Registro de riesgos	20
Tabla 5 Caso de uso No.1	24
Tabla 6 Caso de uso No.2	25
Tabla 7 Caso de uso No.3	26
Tabla 8 Caso de uso No.4	27
Tabla 9 Caso de uso No.5	28
Tabla 10 Caso de uso No.6	29
Tabla 11 Caso de uso No.7	30
Tabla 12 Caso de uso No.8	31
Tabla 13 Caso de uso No.9	32
Tabla 14 Caso de uso No.10	33
Tabla 15 Caso de uso No.11	34
Tabla 16 Prueba No. 1	48
Tabla 17 Prueba No. 2	48
Tabla 18 Prueba No. 3	48
Tabla 19 Prueba No. 4	49
Tabla 20 Prueba No. 5	49
Tabla 21 Prueba No. 6	49
Tabla 22 Prueba No. 7	50
Tabla 23 Prueba No. 8	50

Índice de ecuaciones

Ecuación 1 Punto de función.....	15
Ecuación 2 Kilo-líneas de código	16
Ecuación 3 Esfuerzo aplicado persona-mes.....	17
Ecuación 4 Tiempo de desarrollo en meses	18
Ecuación 5 Productividad	18
Ecuación 6 Personal promedio.....	18

1. Introducción

1.1 Descripción general del problema

Usualmente cuando se comienza a aprender o se refuerzan conocimientos en algún ámbito académico resulta de mucha ayuda tener ejemplos de referencia, y la programación sea cual sea el lenguaje no es una excepción. Es entonces, cuando este software adquiere importancia; pues su principal objetivo es brindar una herramienta entre profesor y alumnos que permita la difusión de ejemplos prácticos (es decir, códigos de programación) de manera sencilla.

1.2 Objetivos

Se propone la creación de un sistema web que le brinde a un profesor la herramienta para difundir software entre sus alumnos, tal que:

- La herramienta sea ordenada
- Brinde las opciones necesarias según el lenguaje de programación
- La información se gestione mediante archivos XML
- Permita la retroalimentación entre profesor y alumno
- Permita al profesor la administración de los alumnos y códigos publicados
- Sea fácil de usar

1.3 Estructura de la memoria

La memoria es un documento escrito donde se realiza un seguimiento del proyecto de forma detallada desde el nacimiento de la idea hasta la obtención de la aplicación final. Los capítulos que aparecerán a lo largo de esta memoria se detallan a continuación.

2. Análisis del Sistema

2.1 Objeto

2.2.1 Generalidades

Existen diversas páginas web que permiten compartir códigos, comentarlos e incluso evaluarlos. Un programador puede subir un fragmento de código y obtener retroalimentación de toda una comunidad online; haciendo más fácil la corrección de errores, el desarrollo colaborativo de software, el aprendizaje de la programación en general, etc. Ejemplo de esto son:

- BitBucket

Es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados. El servicio está escrito en Python.

- GitHub

Forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

- Allura

Software de código abierto para la gestión de repositorios de código, informes de errores, debates, páginas wiki, blogs, etc. Desarrollada en Python.

- Chiliproject

Este es un sistema de gestión de proyectos basado en web. Apoya a tu equipo a lo largo del ciclo de vida del proyecto, desde la creación y discusión de un plan de proyecto, cuestiones de seguimiento y presentación de informes del progreso del trabajo. Aplicación escrita usando Ruby on Rails.

- Stackoverflow

Sitio web desarrollado por Jeff Atwood, este sitio web es utilizado por una comunidad de desarrolladores informáticos, en la cual otros desarrolladores pueden encontrar soluciones a problemas de programación en diferentes lenguajes.

- Kiln

Provee acceso a repositorios via SSH o HTTPS. A;ade un capa adicional de seguridad y limita el acceso a proyectos privados mediante control de acceso basado en usuarios.

- Gitorious

Además del alojamiento de código fuente, Gitorious proporciona a los proyectos: wikis, una interfaz web para solicitudes de fusión o revisión de código y líneas temporales para proyectos o desarrolladores. Gitorious AS distribuye el software de Gitorious bajo la licencia AGPL v3.

- CodePlane

Es una versión menor de GitHub, pero de pago.

- GitLab

Aplicación opensource que nos permite administrar repositorios en git mediante una interfaz web. Podemos administrar; usuarios, repositorios, ramas, permisos, ver los commits, ver el código, mostrar estadísticas y muchas más características.

2.2.2 Perfiles de usuarios

2.2.2.1 Jerarquía de usuarios

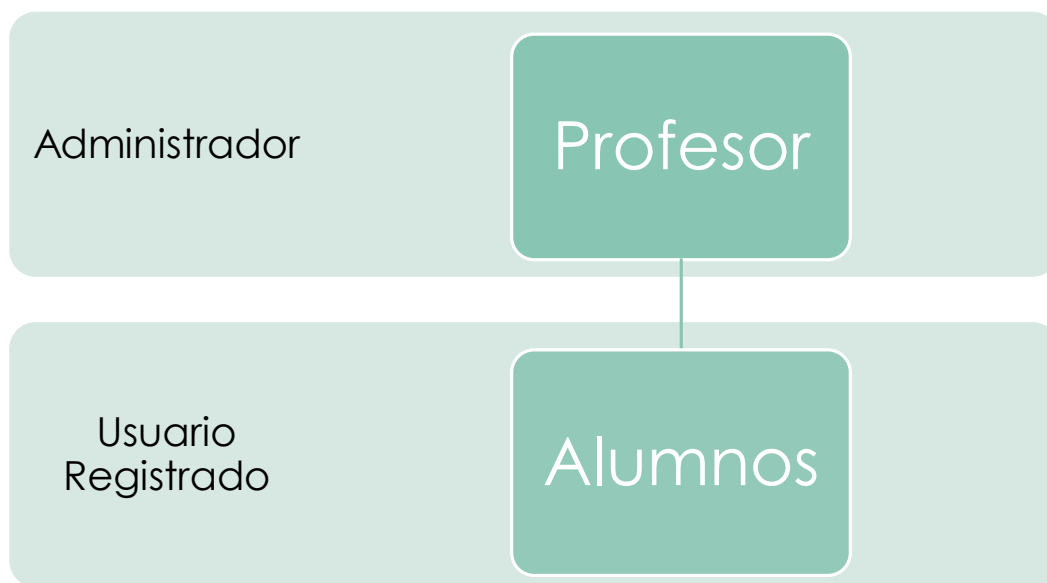


Ilustración 1 Jerarquía de usuarios

2.2.2.2 Profesor

Es el único usuario con perfil de administrador, desempeña diversas actividades; entre ellas la principal: publicar códigos. Gestiono códigos ya publicados y a los demás usuarios, pudiendo incluso eliminar otros usuarios. Restringido a 1 solo usuario, con clave de acceso previamente definida. La eliminación de este perfil no está permitida pues resultaría en la inutilidad del sistema.

2.2.2.3 Alumno

Su principal función en visualizar códigos y comentarlos, es el perfil con mayor número de usuarios. Se puede dar de alta, pero no de baja en el sistema.

2.2.3 Objetivos

2.2.3.1 Objetivo General

El proyecto debe permitir a un usuario administrador (en este caso el profesor) publicar diversos códigos para que los demás usuarios (los alumnos) puedan visualizar el código. Además de proporcionar al profesor control sobre los alumnos y sus grupos.

2.2.3.2 Objetivos específicos

I. Publicación de códigos

Debe permitir al profesor publicar códigos, los cuales deberán corresponder a una sintaxis específica según el lenguaje de programación. El publicar un código implica la escritura de este, la cual se realiza desde el sistema; además de contar con autocompletado de palabras para facilitar su escritura.

II. Registro de alumnos

Un alumno puede registrarse en el sistema, proporcionando información la cual es validada.

III. Recepción de comentarios

Cada alumno registrado es capaz de comentar solo una vez cada código publicado por el profesor. Esto desde luego implica poder ver un listado de los códigos disponibles.

IV. Actividades de control para el profesor

Dichas actividades se dividen en dos:

i. Control sobre los códigos

El profesor es capaz de modificar una línea (o más) de cada código, es capaz de ver un listado de códigos, es capaz de eliminar un código.

ii. Control sobre los alumnos

El profesor puede ver un listado de los grupos y de sus respectivos alumnos, además de eliminar a un alumno de un grupo. Puede calificar del 1-5 los comentarios hechos por los alumnos y si lo considera marcarlo como destacado.

2.2 Funcionalidad

2.2.1 Descripción

El sistema comienza a funcionar cuando los alumnos (usuarios del sistema) se registran, a su vez el profesor (administrador y cuenta principal) debe publicar códigos y administrar a los alumnos.

Una vez que un alumno ha sido dado de alta en el sistema podrá observar y comentar los códigos compartidos por el profesor, además podrá visualizar los comentarios (y sus respectivas evaluaciones) hechos por los demás alumnos.

El profesor puede calificar los comentarios de los alumnos, y administrar los alumnos en el sistema. Es decir, es capaz de organizar a los alumnos en grupos existentes o recién creados. Además de crear códigos los cuales según el lenguaje de programación al que pertenecen son resaltados según su sintaxis y durante su redacción se ofrece la opción de autocompletar según la sentencia.

2.3.2 Paradigma

El proyecto será desarrollado siguiendo el modelo en cascada:

- El proyecto es "sencillo", no es proyecto complejo por lo que no es necesario un equipo de trabajo grande. Además se sabe cómo abordar la problemática planteada.

- Se tiene bien definido lo que el proyecto debe hacer, no se esperan cambios mayores a futuro.

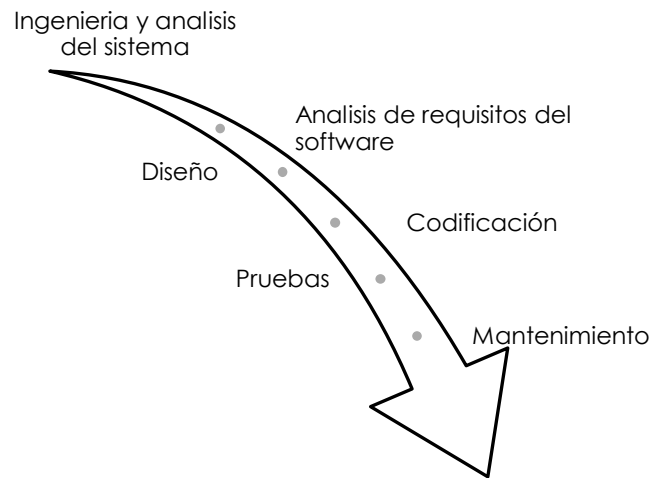


Ilustración 2 Modelo en Cascada

2.2.2 Recursos

Con el fin de poder desarrollar este proyecto y conocer su viabilidad, es necesario contemplar la necesidad de disponer de diferentes tipos de recursos. Estos recursos los podemos dividir en tres grupos: hardware, software y recursos humanos.

Respecto a los usuarios, sólo serán necesarios conocimientos de informática a nivel de usuario, ya que el entorno de la aplicación será sencillo pero a la vez funcional. Se realizará una formación para facilitar el aprendizaje y sacar el máximo provecho de la aplicación, esta formación será realizada por el autor del proyecto y su coste ha sido contabilizado dentro de este estudio de viabilidad.

2.2.2.1 Hardware

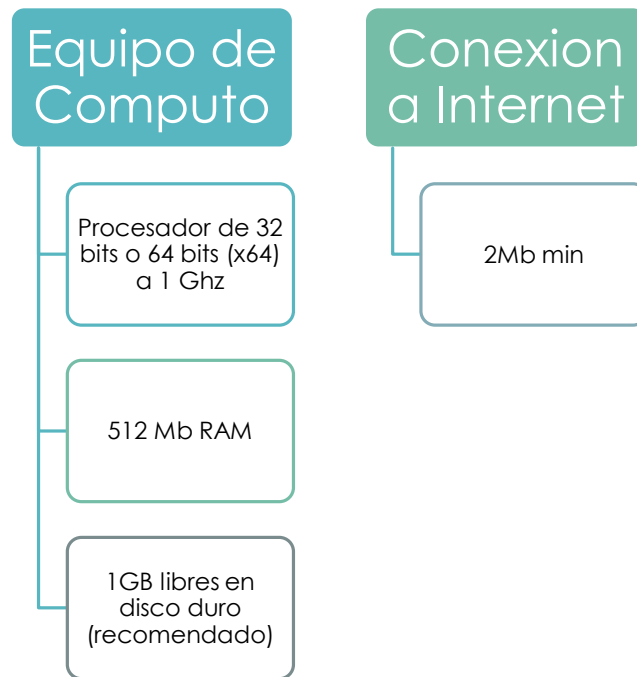


Ilustración 3 Recursos de Hardware

2.2.2.2 Recursos lógicos

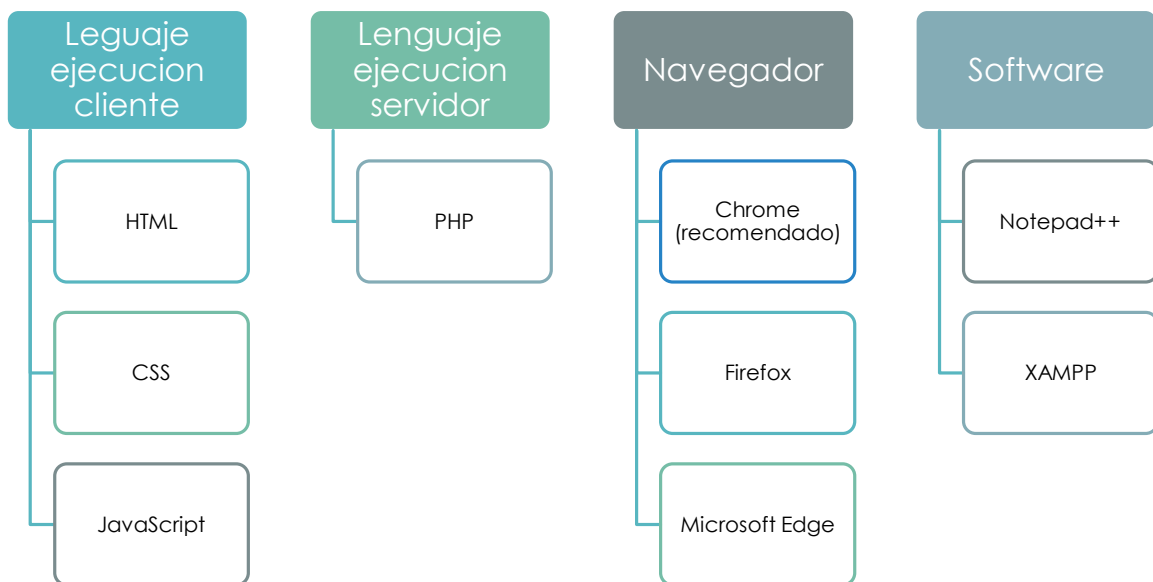


Ilustración 4 Recursos lógicos

2.2.2.3 Recursos Humanos

Analista	Programador	Tester	Diseñador Web
<ul style="list-style-type: none"> •Capacidad para resolución de problemas •Evaluación del software •Supervisión del desarrollo del software 	<ul style="list-style-type: none"> •Conocimientos en desarrollo de sistemas web •HTML •PHP •Javascript •DOM •AJAX 	<ul style="list-style-type: none"> •Conocimiento en el flujo de sistemas Web •Retroalimentación de información relativa a la funcionalidad del software 	<ul style="list-style-type: none"> •CSS •Diseño de MockUps

Ilustración 5 Recursos humanos

2.4 Métricas y estimaciones

2.4.1 Valores de Ajuste de Compatibilidad

i	Pregunta	Respuesta	Valor de Ajuste de Compatibilidad
1	¿Requiere el sistema copias de seguridad y de recuperación confiables?	No necesariamente, es un repositorio de códigos de programación, sin embargo la información que maneja no es de alta importancia.	Influencia Mínima (1)
2	¿Se requiere comunicación de datos?	Si. Los usuarios deben ingresar.	Influencia Significativa (4)
3	¿Existen funciones de procesamiento distribuido?	No. El sistema no maneja una red de computadoras.	Sin Influencia (0)
4	¿Es crítico el rendimiento es decir es crucial el desempeño?	Si, ya que será usado por varios grupos de usuarios.	Influencia Apreciable (3)
5	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	Si. El sistema será ejecutado en un explorador.	Influencia Significativa (4)

6	¿Requiere el sistema entrada de datos en línea?	Sí, todos los usuarios ingresan datos en algún punto.	Influencia Muy Fuerte (5)
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	No. Cada usuario interactúa de forma individual.	Sin Influencia (0)
8	¿Se actualizan los ALI en línea?	Si, el sistema se modifica en línea.	Influencia Significativa (4)
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	No, los datos no son complejos en términos del sistema.	Influencia Mínima (1)
10	¿Es complejo el procesamiento interno?	No, las operaciones no son complejas.	Influencia Mínima (1)
11	¿Se ha diseñado el código para ser reutilizable?	No, no se ha planeado.	Influencia Mínima (1)
12	¿Están incluidas en el diseño la conversión y la instalación'?	Es un sistema web, no requiere una instalación como tal.	Influencia Mínima (1)
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	No, el sistema está pensado para un usuario "profesor" en específico.	Sin Influencia (0)
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	Si, muchos usuarios interactuaran con el sistema. Es importante la facilidad del funcionamiento.	Influencia Muy Fuerte (5)
ΣFi			30

2.4.2 Puntos de función

- **Entradas de usuario:**
 - **Entradas de alumnos:**
 - Registro
 - Comentarios
 - **Entradas de profesor:**
 - Creación y gestión de código
- **Salidas de usuario:**
 - Salidas profesor
 - Códigos
- **Peticiones de usuario**
 - Peticiones profesor
 - Ver y gestionar Códigos
 - Ver y gestionar grupos
 - Peticiones alumno
 - Ver códigos
- **Archivos**
 - Códigos publicados

Puntos de Función	Cuenta	Factor de Ponderación	
Entradas de usuario	2	Simple (×3)	6
Salidas de usuario	1	Simple (×4)	4
Peticiones de usuario	3	Simple (×3)	9
Archivos	1	Simple (×7)	7
Numero de interfaces externas	0	Simple (×5)	0
		Cuenta Total	28

$$PF = \text{cuenta-total} \times (0,65 + 0,01 \times \sum Fi)$$

$$PF = 28 \times (0,65 + 0,01 \times 30)$$

$$PF = 39 \times 1,14$$

$$PF = 26.6$$

Ecuación 1 Punto de función

2.4.3 Método (COCOMO)

2.4.3.1 KDLC (Kilo-líneas de código).

$$KLDC = \frac{(PF * \text{Líneas de código por cada PF})}{1000}$$

$$KLDC = \frac{(26.6 * 150)}{1000}$$

$$KLDC = 3.99$$

Ecuación 2 Kilo-líneas de código

Nota: Líneas de código por cada PF es según el lenguaje a usar mayoritariamente, en este caso PHP.

2.4.3.2 Obtención de FAE.

Conductores de Coste	Justificación	Valoración
Fiabilidad requerida del software	El software es un repositorio, sin embargo la información no es de alta importancia.	Normal (1.00)
Tamaño de la base de datos	La información se gestionara con archivos XML, no se usara una base de datos.	Bajo (0.87)
Complejidad del producto	No es un sistema muy complejo.	Normal (1.00)
Restricciones del tiempo de ejecución	No tiene restricciones en el tiempo de ejecución, pero si un orden.	Normal (1.00)
Restricciones del almacenamiento principal	Sin mayores restricciones.	Normal (1.00)
Volatilidad de la máquina virtual	Al ser un sistema web, la carga es baja.	Bajo (0.87)
Tiempo de respuesta del ordenador	No es un software muy complejo, solo requiere de un navegador web.	Normal (1.00)
Capacidad del analista	El sistema no es complejo.	Bajo (0.87)
Experiencia en la aplicación	Es un sistema web, no requiere conocimientos específicos.	Normal (1.00)
Capacidad de los programadores	El sistema es sencillo, no requiere conocimientos específicos.	Bajo (0.87)
Experiencia en SO utilizado	El sistema requiere familiarizaron con el SO, pero	Normal (1.00)

	no conocimientos especializados.	
Experiencia en el lenguaje de programación	Requiere conocimientos en php, javascript y AJAX.	Normal (1.00)
Prácticas de programación modernas	Se utilizara php.	Normal (1.00)
Utilización de herramientas de software	No es de gran complejidad.	Normal (1.00)
Limitaciones de planificación del proyecto	El proyecto no cuenta con limitaciones mayores.	Normal (1.00)
FAE		0.5729

2.4.3.3 Tipo de proyecto.

El software no tiene restricciones "rígidas" con respecto al procesador e interfaces de software, el grupo que lo desarrollara tiene experiencia intermedia.

Tipo de Proyecto	a	b	c	d
Orgánico	2.40	1.05	2.5	0.38

2.4.4 Ecuaciones del modelo COCOMO.

2.4.4.1 Esfuerzo aplicado persona-mes

$$E = (a)(KLDC)^b FAE$$

$$E = (2.4)(3.99)^{1.05}(0.5729)$$

$$\mathbf{E = 5.88}$$

Ecuación 3 Esfuerzo aplicado persona-mes

2.4.4.2 Tiempo de desarrollo en meses

$$D = (c)(E)^d$$

$$D = (2.5)(5.88)^{0.38}$$

$$\mathbf{D = 4.9}$$

Ecuación 4 Tiempo de desarrollo en meses

2.4.4.3 Productividad

$$PR = \frac{1000 \times KLCD}{E}$$

$$PR = \frac{1000 \times 3.99}{5.88}$$

$$\mathbf{PR = 678.57}$$

Ecuación 5 Productividad

2.4.4.4 Personal promedio

$$P = \frac{E}{D}$$

$$P = \frac{5.88}{4.9}$$

$$\mathbf{P = 1.2}$$

Ecuación 6 Personal promedio

2.5 Costes

En este análisis se contemplan los gastos materiales necesarios para la implantación del software.

2.5.1 Costes Hardware

Recurso	Costo Mínimo	Costo Máximo	periodicidad
Computadora	1,300 pesos	2,500 pesos	única
Acceso a Internet	0 pesos	700 pesos	mensual

Tabla 1 Costes Hardware

2.5.2 Costes Recursos Lógicos

Recurso	Costo Mínimo	Costo Máximo	periodicidad
Lenguaje ejecución cliente	No aplica	No aplica	única
Lenguaje ejecución servidor	No aplica	No aplica	única
Navegador Web	No aplica (Software libre)	No aplica (Software libre)	única
Software	No aplica (Software libre)	No aplica (Software libre)	única

Tabla 2 Costes recursos lógicos

2.5.3 Costes Recursos Humanos

Recurso	Costo Mínimo	Costo Máximo	periodicidad
Analista de software	18, 000 pesos	23, 000 pesos	Mensual (recurrente)
Programador	10, 000 pesos	15, 000 pesos	Mensual (recurrente)
Tester	1, 500 pesos	2, 000 pesos	Mensual (ocasional)
Diseñador Web	5, 000 pesos	7, 000 pesos	Mensual (ocasional)

Tabla 3 Costes recursos humanos

2.6 Evaluación de riesgos

- Inexactitud de los flujos de efectivo del proyecto.

Dependerá de los recursos, si son calculados principalmente mediante los requisitos del cliente se calcularán con exactitud. En caso contrario, si se adoptan cambios en la estructura es posible caer en la inexactitud.

- Aspectos humanos

Depende del equipo de trabajo y la capacidad de adaptación, al igual que aspectos exteriores que pueden implicar el abandono de algún integrante. La reintegración de un elemento tendría repercusiones en todo el proyecto.

- El manejo del cronograma

Los tiempos pueden exceder debido a características que no se habían contemplado con anterioridad, a un mal análisis o información omitida por el cliente, lo que desfasaría al equipo y al proyecto en general.

2.6.1 Registro de riesgos

Riesgo	Probabilidad de pérdida	Magnitud de la pérdida (semanas)	Exposición a riesgo
Planificación demasiado optimista	50%	6	50%
Los recursos no están disponibles en su momento	10%	2	10%
Diseño inadecuado (hay que volver a diseñar)	30%	4	40%
Necesite más espacio en servidor.	10%	½	0
Inexactitud de los flujos de efectivo del proyecto	80%	Indefinido	10%
Aspectos humanos	30%	3	50%
El manejo del cronograma	35%	2	50%

Tabla 4 Registro de riesgos

2.7 Planeación del proyecto

El método cocomo ha arrojado un estimado de $4.9 \approx 5$ meses para 1.2 personas, es decir 1 persona deberá realizar el proyecto en 5 meses. Tomando esta información en conjunto con el registro de riesgos se planearan los tiempos estimados según las especificaciones del proyecto haciendo uso del diagrama de Gantt.

2.7.1 Diagrama de Gantt

		2015												2016							
		Octubre				Noviembre				Diciembre				Enero				Febrero			
No.	Actividad	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Análisis del sistema																				
	Definición de perfiles usuario																				
	Definición de objetivos																				
	Métricas y estimaciones																				
	Costos y riesgos																				
2	Requisitos del software																				
	Requisitos funcionales																				
	Requisitos no funcionales																				
	Modelado de casos de uso																				
3	Diseño																				
	Arquitectura del sitio web																				
	Estructura de archivos XML																				
	Interfaz grafica																				
4	Codificación																				
5	Pruebas																				
6	Mantenimiento																				

Ilustración 6 Diagrama de Gantt

3. Análisis de Requisitos

3.1 Introducción

En esta sección, se lleva a cabo un estudio preciso enfocado a la obtención de un buen resultado de la aplicación y poder pronosticar los posibles fallos, es necesaria una fase en la que se analizan los requisitos del software. Esta fase es conocida como Análisis de requisitos y nos ayudará a comprender, cual es el problema o necesidad que se pretende solucionar y como se afrontará. Es decir, que ha de hacer el sistema, o que necesidad resolverá.

3.2 Requisitos funcionales

Los requisitos funcionales son aquellos que especifican cada funcionalidad del sistema, además de indicar como se ha de comportar la aplicación.

El núcleo del requisito es la descripción del comportamiento del requisito, que ha de ser clara y concisa. Este comportamiento puede proveer las reglas empresariales o de negocio, o bien pueden ser descubiertas por las conversaciones con los clientes.

3.2.1 RF1: Restricciones tipo de cuenta

Las diferencias y restricciones según el tipo de cuenta que el usuario tenga debe estar bien definidas, es decir solo el usuario administrador debe tener acceso a las funciones predestinadas.

3.2.2 RF2: Creación y publicación de contenidos

Debe permitir que el proceso de creación sea sencillo de usar, además debe brindar las herramientas necesarias para la publicación correcta de contenidos (comentarios y códigos).

3.2.3 RF3: Gestión de usuarios

El sistema deberá permitir el registro y eliminación de usuarios, garantizando la existencia de un administrador (profesor) en todo momento.

3.3 Requisitos no funcionales

Los requisitos no funcionales del proyecto son las restricciones impuestas por el cliente o que son inherentes a las características de la aplicación. Estas restricciones afectan al diseño del propio sistema y tienen relación directa con el rendimiento de la solución desarrollada.

3.3.1 Requisitos de rendimiento

3.3.1.1 RNF1: Concurrencia

El software deberá soportar la visita de varios usuarios al mismo tiempo, sin embargo no se esperan visitas masivas.

3.3.1.2 RNF2: Tiempo de respuesta

El tiempo de respuesta del sitio web debe ser aceptable, debe adaptarse al hardware que los usuarios usen.

3.3.2 Requisitos de interfaz

3.3.2.1 RNF3: Facilidad de uso

El sistema debe ser sencillo de usar para cualquier perfil de usuario, no debe suponer complicaciones mayores para su entendimiento.

3.4 Modelado de casos de uso

3.4.1 Introducción

Los diagramas de casos de uso describen lo que realiza un sistema desde el punto de vista del observador externo, priorizando el qué más que el cómo. Se plantean escenarios diferentes para saber qué pasará cuando un usuario interactúa con el sistema, proporcionando un resumen de la tarea.

3.4.2 Casos de uso de la aplicación

3.4.2.1 Diagrama caso de uso registro de usuario nuevo en el sistema

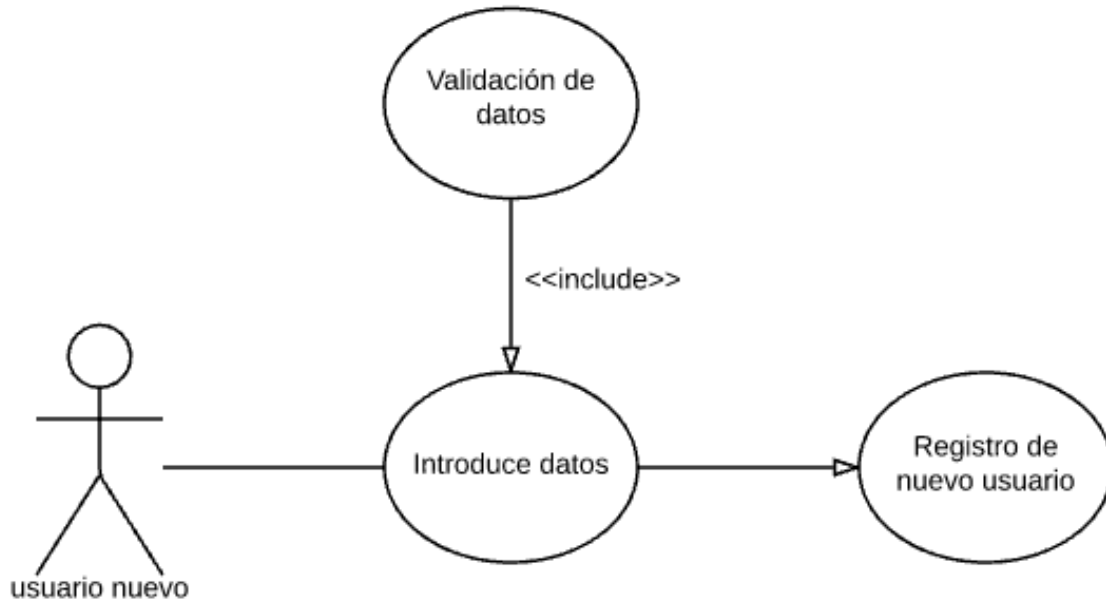


Ilustración 7 Caso de uso No.1

3.4.2.1.1 Descripción caso de uso registro de usuario nuevo en el sistema

Actores involucrados	Usuario registrado
Descripción	El usuario ingresa al sistema
Flujo básico	El usuario ingresa sus datos (nombre y contraseña), el sistema lo registra en los archivos.
Pre condiciones	Los datos proporcionados no deben existir en los archivos correspondientes.
Post condiciones	El usuario queda registrado, el sistema muestra un mensaje para informar.

Tabla 5 Caso de uso No.1

3.4.2.2 Diagrama caso de uso ingreso al sistema

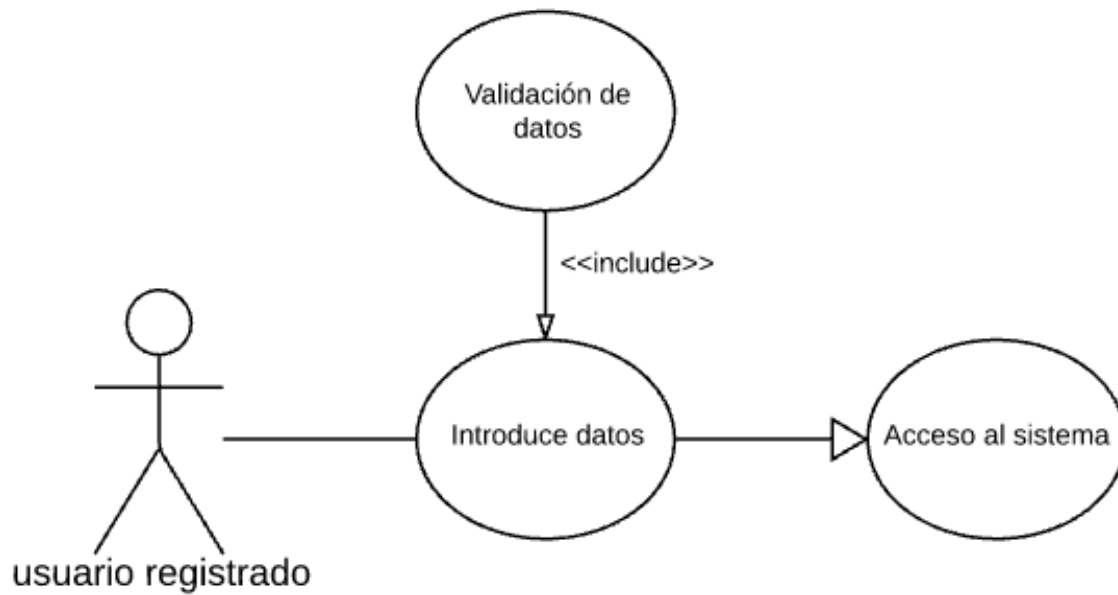


Ilustración 8 Caso de uso No.2

3.4.2.2.1 Descripción caso de uso ingreso al sistema

Actores involucrados	Usuario registrado
Descripción	El usuario ingresa al sistema
Flujo básico	El usuario ingresa sus datos (nombre y contraseña), el sistema valida la información y se da acceso al sistema.
Pre condiciones	El usuario debe haberse registrado con anterioridad.
Post condiciones	El usuario es llevado al menú principal correspondiente.

Tabla 6 Caso de uso No.2

3.4.2.2 Diagrama de caso de uso menú principal alumno

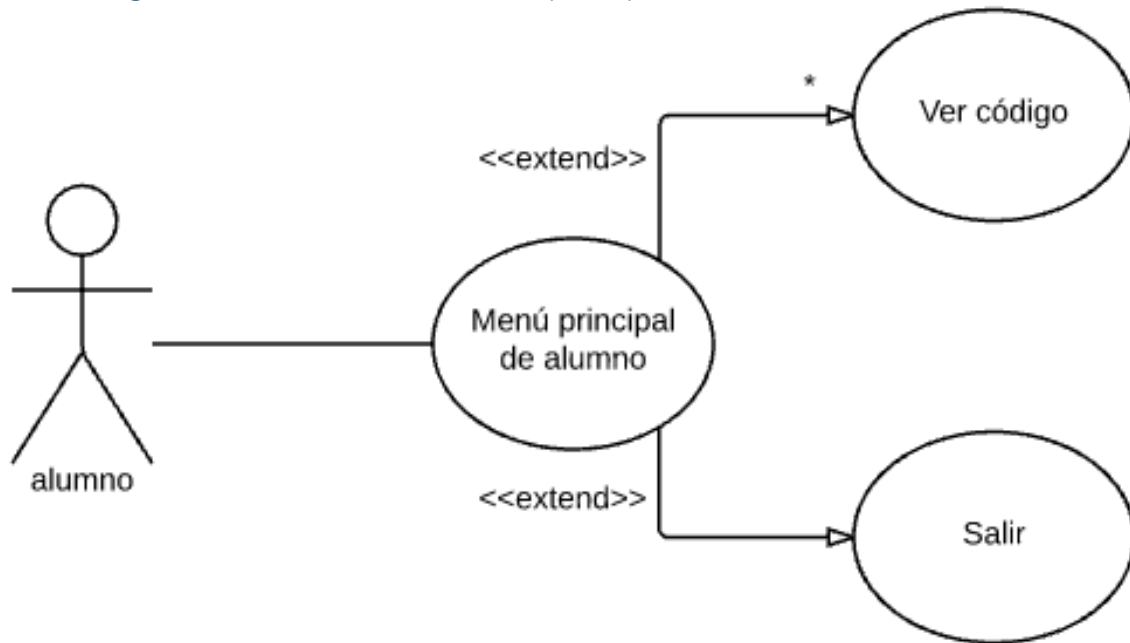


Ilustración 9 Caso de uso No.3

3.4.2.2.1 Descripción caso de uso menú principal alumno

Actores involucrados	Usuario registrado (perfil: alumno)
Descripción	El usuario navega el menú principal correspondiente al perfil de alumno.
Flujo básico	El usuario se encuentra en el menú principal del perfil de alumno, donde puede elegir salir o ver los códigos disponibles.
Pre condiciones	El usuario debe haberse registrado e ingresado al sistema.
Post condiciones	El usuario navega el menú principal correspondiente al perfil de alumno.

Tabla 7 Caso de uso No.3

3.4.2.3 Diagrama de caso de uso ver código – alumno

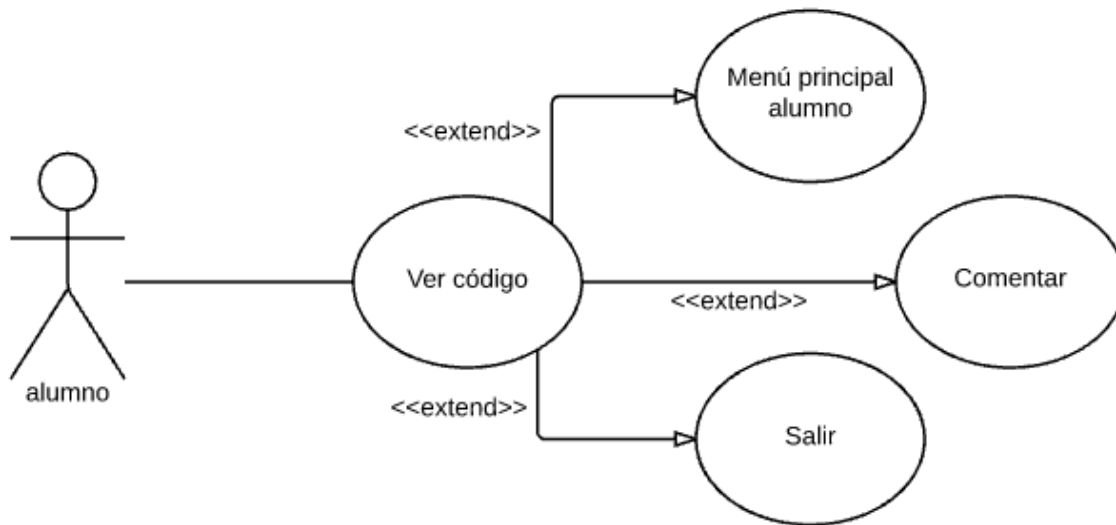


Ilustración 10 Caso de uso No.4

3.4.2.3.1 Descripción de caso de uso ver código – alumno

Actores involucrados	Usuario registrado (perfil: alumno)
Descripción	El usuario ve un código.
Flujo básico	El usuario ingresa a la página de un código específico, donde puede elegir entre: ir al menú principal del perfil de alumno, comentar el código y salir del sistema.
Pre condiciones	El usuario debe haberse registrado e ingresado al sistema, donde deberá elegir un código existente. El usuario con el perfil de Profesor debe haber creado y publicado un código.
Post condiciones	El usuario navega la página de un código ya publicado.

Tabla 8 Caso de uso No.4

3.4.2.3 Diagrama caso de uso comentar – alumno

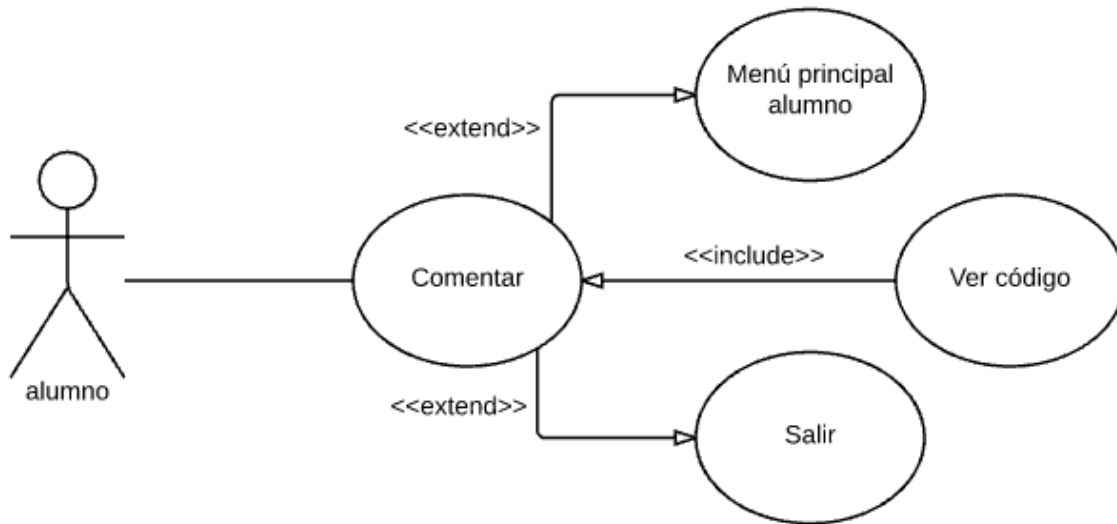


Ilustración 11 Caso de uso No.5

3.4.2.3.1 Descripción caso de uso comentar – alumno

Actores involucrados	Usuario registrado (perfil: alumno)
Descripción	El comenta un código.
Flujo básico	El usuario ingresa su comentario en el cuadro de texto, da click en el botón. El comentario se registra y el sistema informa el resultado de la operación.
Pre condiciones	El usuario debe haberse registrado e ingresado al sistema, donde deberá elegir un código existente. El usuario con el perfil de Profesor debe haber creado y publicado un código.
Post condiciones	El usuario navega la página de un código ya publicado. Donde podrá observar su comentario.

Tabla 9 Caso de uso No.5

3.4.2.3 Diagrama de caso de uso Menú principal – Profesor

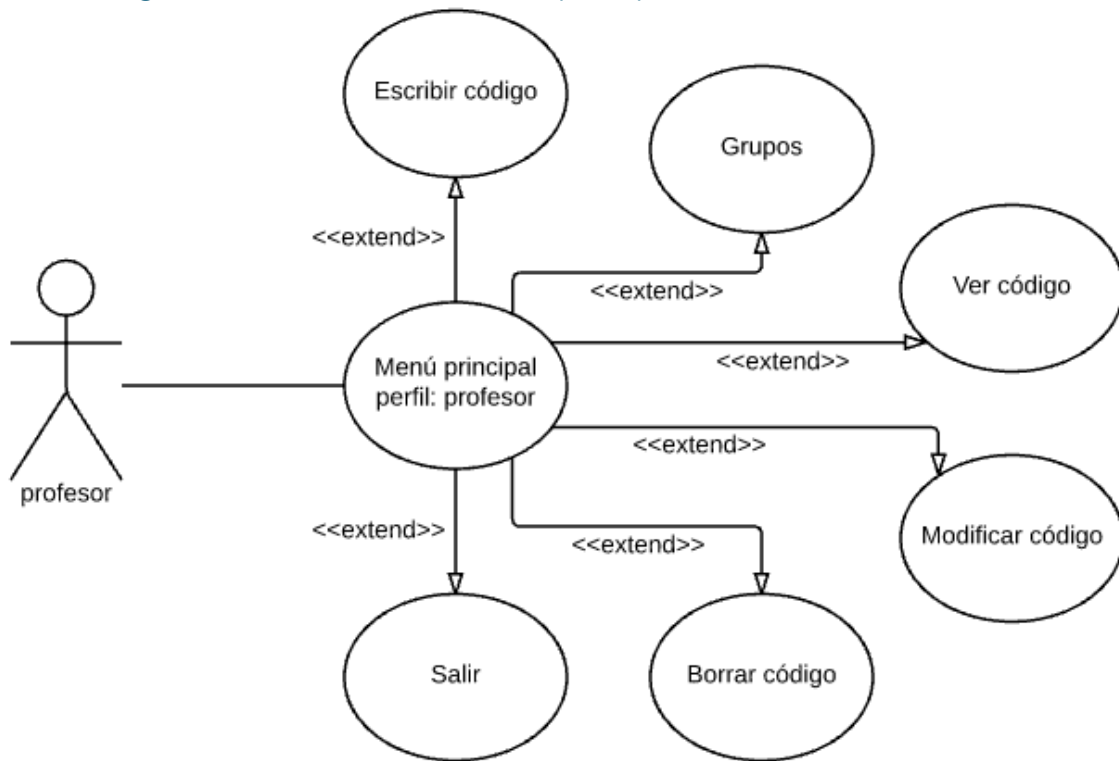


Ilustración 12 Caso de uso No.6

3.4.2.3.1 Descripción de caso de uso Menú principal - Profesor

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario se encuentra en la página del menú principal correspondiente a su perfil.
Flujo básico	El usuario puede elegir entre: escribir código, grupos, ver un código específico, modificar un código específico, borrar un código específico o salir del sistema.
Pre condiciones	El usuario debe haber ingresado al sistema.
Post condiciones	El usuario navega la página del menú principal correspondiente a su perfil.

Tabla 10 Caso de uso No.6

3.4.2.4 Diagrama de caso de uso Escribir código

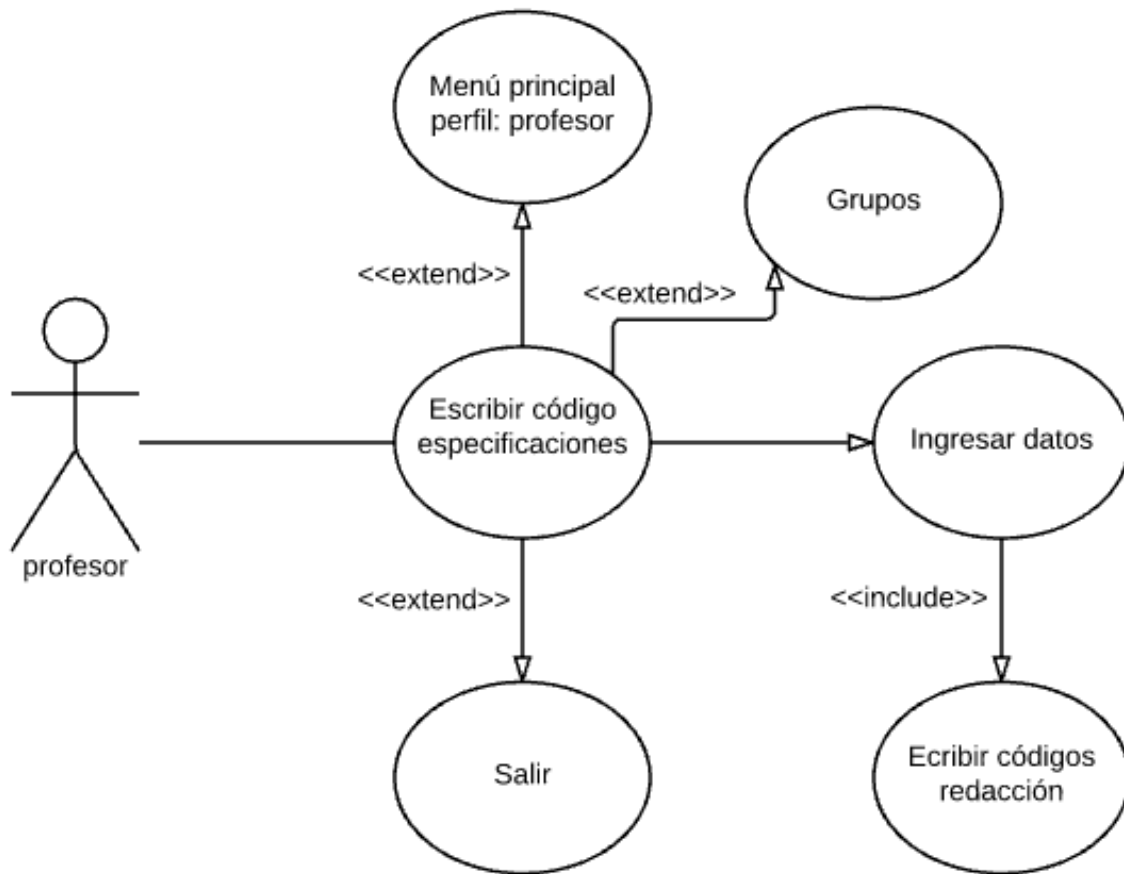


Ilustración 13 Caso de uso No.7

3.4.2.4.1 Descripción de caso de uso Escribir código

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario redacta un código a ser publicado, que tiene ciertas características.
Flujo básico	El usuario puede elegir entre: ir al menú principal, Grupos, Salir o Ingresar los datos requeridos. Si se ingresan los datos requeridos el sistema automáticamente se ira a la redacción del código.
Pre condiciones	El usuario debe haber ingresado al sistema.
Post condiciones	El usuario ingresara los datos requeridos y podrá redactar su código.

Tabla 11 Caso de uso No.7

3.4.2.5 Diagrama de casos de uso Gestión de grupos

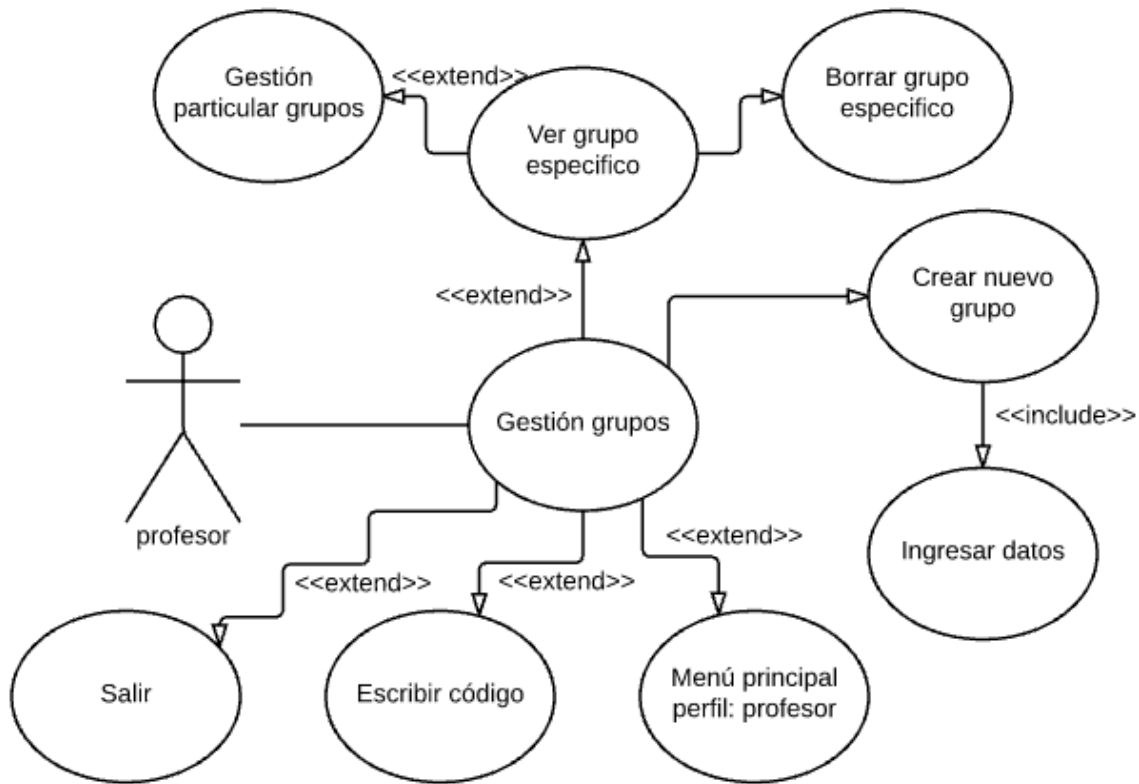


Ilustración 14 Caso de uso No.8

3.4.2.5.1 Descripción de casos de uso Gestión de grupos

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario realiza las actividades relacionadas con la gestión de grupos.
Flujo básico	El usuario puede elegir entre: ir al menú principal, Grupos, Salir, Ver un grupo específico o crear un nuevo grupo. Si se elige la opción de crear un nuevo grupo se deberán ingresar algunos datos.
Pre condiciones	El usuario debe haber ingresado al sistema.
Post condiciones	El usuario ingresara los datos requeridos.

Tabla 12 Caso de uso No.8

3.4.2.6 Diagrama de casos de uso de Gestión particular grupo

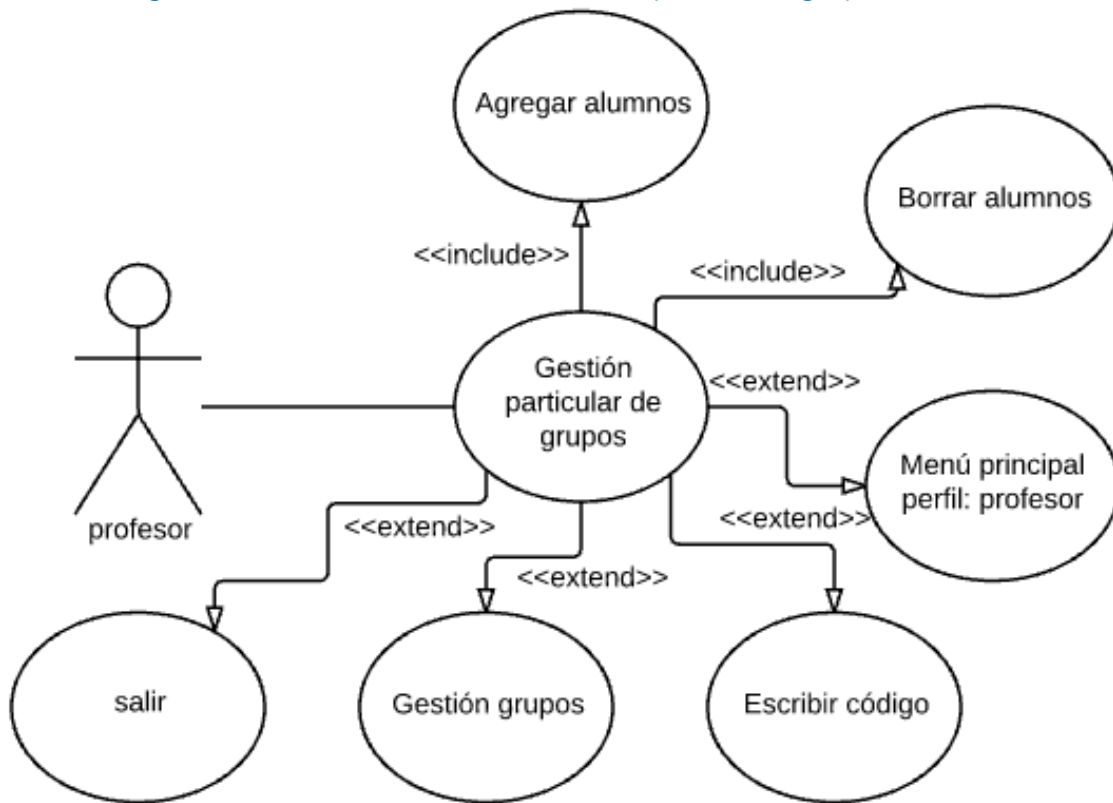


Ilustración 15 Caso de uso No.9

3.4.2.6.1 Descripción de casos de uso de Gestión particular grupo

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario realiza las actividades relacionadas la gestión de un grupo.
Flujo básico	El usuario puede elegir entre: ir al menú principal, escribir código, Grupos, Salir, agregar alumnos, eliminar alumnos.
Pre condiciones	El usuario debe haber ingresado al sistema, debe existir al menos un grupo.
Post condiciones	El usuario modificara u observara un grupo en específico.

Tabla 13 Caso de uso No.9

3.4.2.7 Diagrama de casos de uso Ver Código

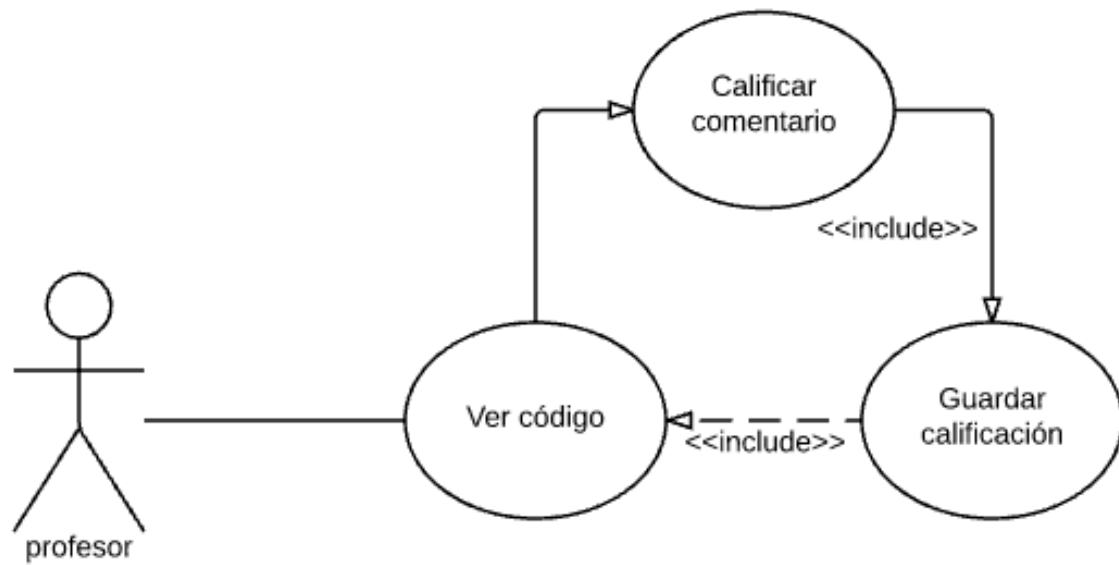


Ilustración 16 Caso de uso No.10

3.4.2.7.1 Descripción de casos de uso Ver Código

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario visualiza un código el cual puede o no tener comentarios, los cuales podrá calificar.
Flujo básico	El usuario ve un código, lee los comentarios, evalúa o no los comentarios.
Pre condiciones	El usuario debe haber ingresado al sistema, debe de existir al menos un código publicado.
Post condiciones	El usuario habrá observado un código y sus respectivos comentarios.

Tabla 14 Caso de uso No.10

3.4.2.8 Diagrama de casos de uso Modificar Código

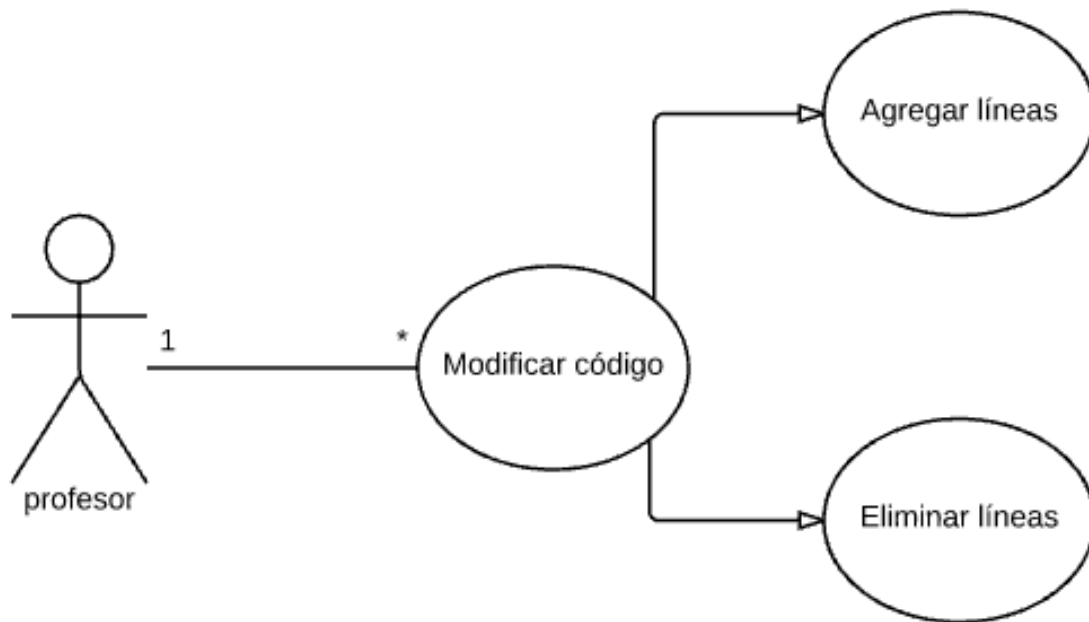


Ilustración 17 Caso de uso No.11

3.4.2.8.1 Descripción de casos de uso Modificar Código

Actores involucrados	Usuario administrador (perfil: profesor)
Descripción	El usuario modifica un código previamente publicado.
Flujo básico	El usuario agrega o elimina líneas de código, guarda los cambios.
Pre condiciones	El usuario debe haber ingresado al sistema, debe de existir al menos un código publicado.
Post condiciones	El usuario modifico un código.

Tabla 15 Caso de uso No.11

4. Diseño del Proyecto

4.1 Introducción

Este capítulo incluye el documento de diseño del sistema. Este documento permitirá tanto al analista como al usuario formarse una idea de la aplicación final. Se describen todos los aspectos de la aplicación necesarios en el momento de desarrollar la aplicación, exponiendo las funcionalidades del sistema más detalladamente, a nivel de módulo, y realizando una descripción de su aspecto final.

4.2 Arquitectura del sitio web

El sitio seguirá una organización *lógica mixta*, la estructura del sistema de archivos propuesta es la siguiente:

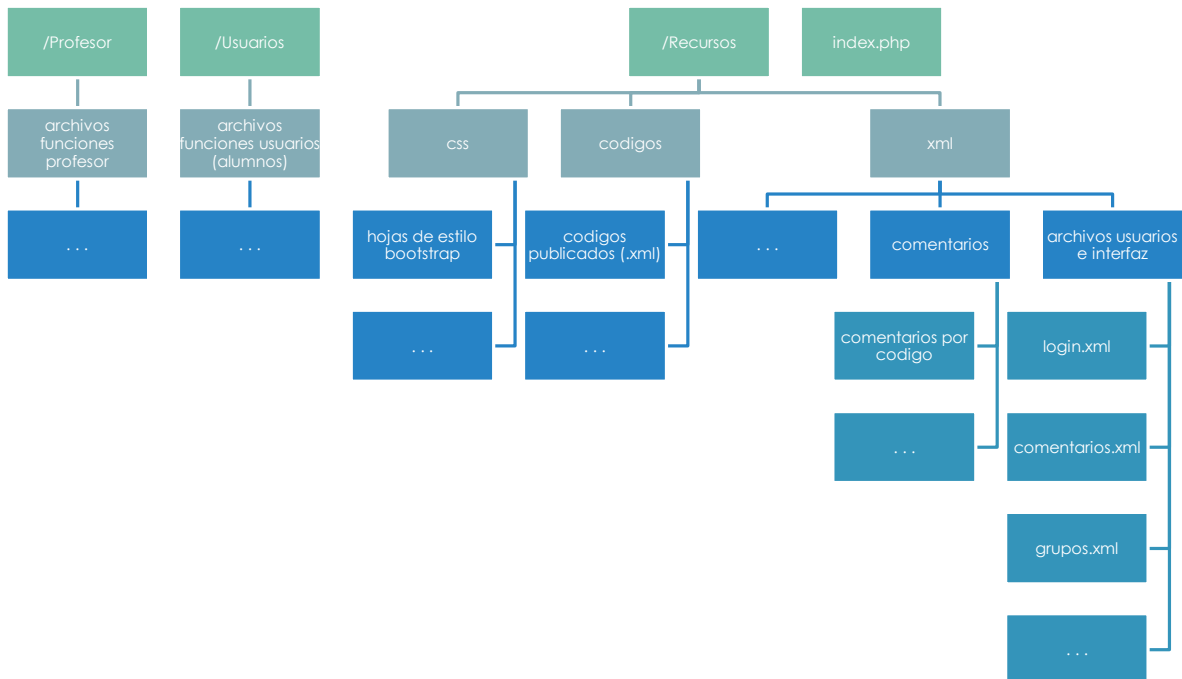


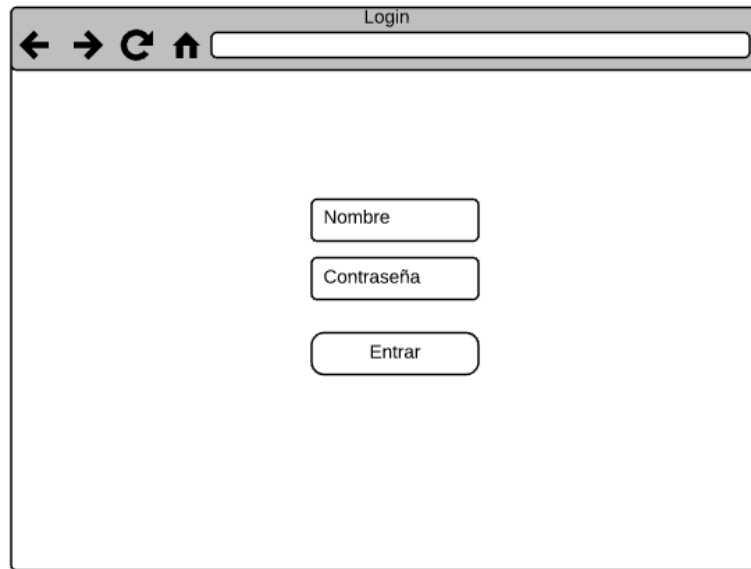
Ilustración 18 Arquitectura de datos

4.3 Interfaz grafica

4.3.1 Mock-Ups

4.3.1.1 Entrar al sitio

En esta ventana el usuario podrá entrar al sitio ingresando sus datos, es usada por todos los perfiles de usuarios.

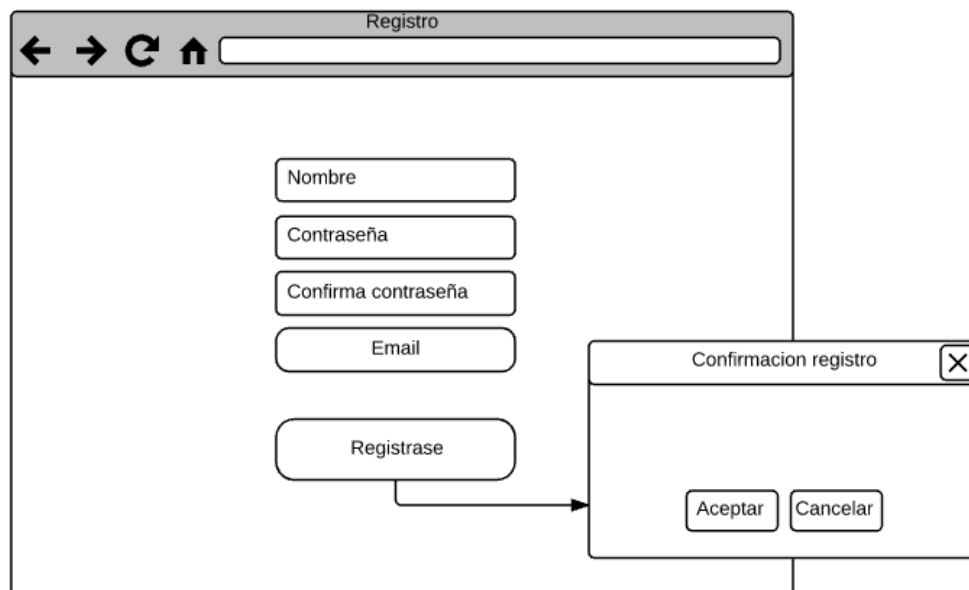


The mockup shows a web browser window titled "Login". The address bar is empty. The main content area contains three input fields stacked vertically: "Nombre", "Contraseña", and "Entrar".

Ilustración 19 Mock up No. 1

4.3.1.2 Registrarse

Página en la que el usuario (alumno) ingresa los datos necesarios para ingresar al sitio.



The mockup shows a web browser window titled "Registro". The address bar is empty. The main content area contains five input fields stacked vertically: "Nombre", "Contraseña", "Confirma contraseña", "Email", and "Regístrate". A confirmation dialog box titled "Confirmacion registro" is open, showing "Aceptar" and "Cancelar" buttons. An arrow points from the "Regístrate" button to the dialog box.

Ilustración 20 Mock up No. 2

4.3.1.3 Menú principal – alumno

Página a la que se redirige a un usuario con el perfil de alumno tras haber iniciado sesión.

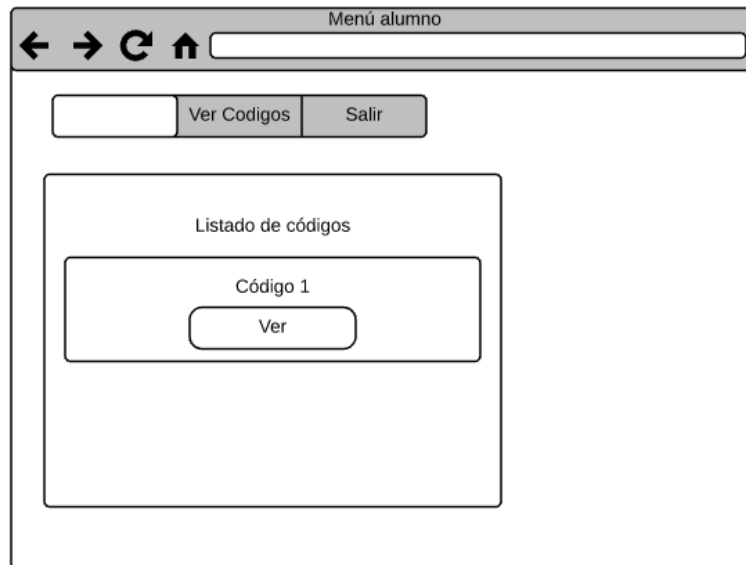


Ilustración 21 Mock up No. 3

4.3.1.4 Ver código – alumno

Desde el menú principal de alumno, se debe seleccionar un código. Aquí se le permite al usuario comentar un código y leer los demás comentarios.

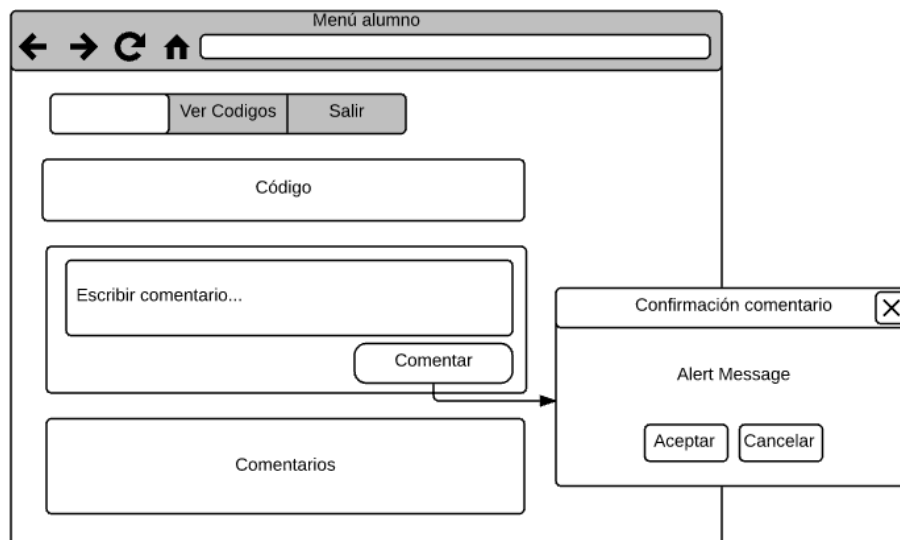


Ilustración 22 Mock up No. 4

4.3.1.5 Menú principal – profesor

Aquí se redirige a un usuario con el perfil de profesor tras haber iniciado sesión.

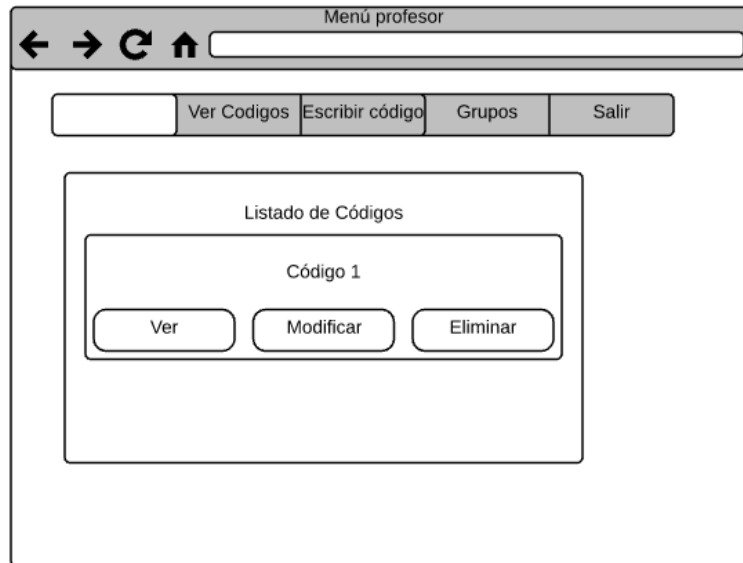


Ilustración 23 Mock up No. 5

4.3.1.6 Ver código – profesor

Tras elegir en menú principal un código y elegir la opción ver se le re direcciona al profesor a esta página. Aquí ve el código y los comentarios los cuales podrá comentar.

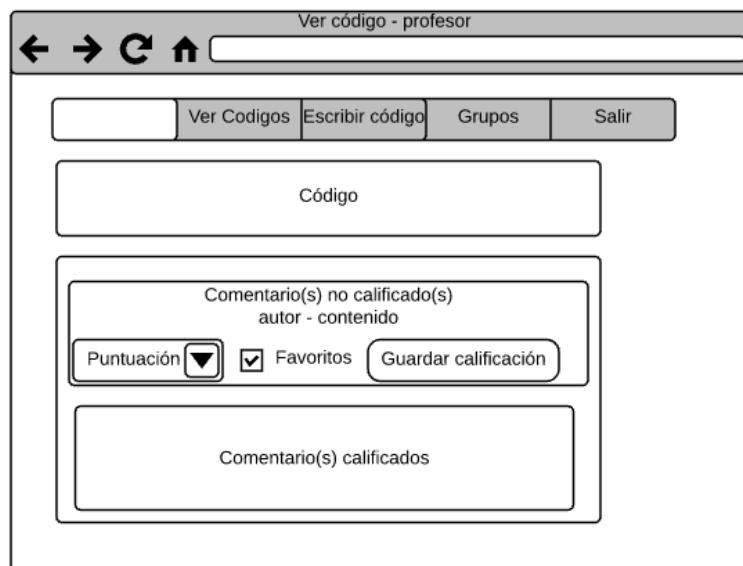


Ilustración 24 Mock up No. 6

4.3.1.7 Modificar código – profesor

Tras elegir en menú principal un código y elegir la opción modificar se le re direcciona al profesor a esta página.

The mockup shows a web browser window titled "Modificar código - profesor". The address bar is empty. Below the browser window, there is a navigation bar with five buttons: "Ver Codigos", "Escribir código", "Grupos", and "Salir". The "Escribir código" button is highlighted. Below the navigation bar, there is a "Guardar" button. Below that, there is a "Sentencia" input field and an "Agregar línea" button. Below that, there is a "Código" input field. Below that, there is a "Borrar ultima línea" button.

Ilustración 25 Mock up No. 7

4.3.1.8 Escribir código – especificaciones

Al elegir en la barra superior la opción Escribir código, se le enviara aquí.

The mockup shows a web browser window titled "Escribir código - especificaciones". The address bar is empty. Below the browser window, there is a navigation bar with five buttons: "Ver Codigos", "Escribir código", "Grupos", and "Salir". The "Escribir código" button is highlighted. Below the navigation bar, there is a "Lenguaje" dropdown menu with a downward arrow. Below that, there is a "Número de columnas ..." input field. Below that, there is a "Número de filas ..." input field. Below that, there is a "Nombre del archivo..." input field. Below that, there is an "Enviar" button.

Ilustración 26 Mock up No. 8

4.3.1.9 Escribir código – redacción

Esta ventana solo se mostrara al elegir enviar en la ventana Escribir código – especificaciones, aquí se redacta el código a publicar.

Ilustración 27 Mock up No. 9

4.3.1.10 Ver Grupos – profesor

Al elegir en la barra superior la opción Grupos, se le enviara aquí. Donde se podrá agregar un grupo.

Ilustración 28 Mock up No. 10

4.3.1.11 Ver grupo – profesor

Aquí se mostraran las opciones para la gestión de un grupo en específico.

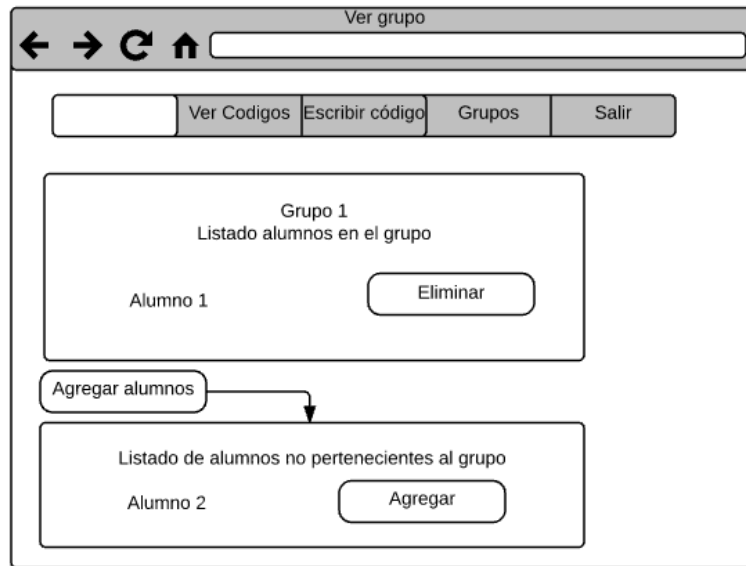


Ilustración 29 Mock up No. 11

5. Codificación

5.1 Introducción

Para la realización de este proyecto se ha trabajado con el programa NotePad++, un editor de texto gratuito que proporciona al usuario muchas posibilidades de trabajo, como la colocación de etiquetas o sintaxis propias del lenguaje de trabajo. Además con el uso de colores diferencia el uso de diversos lenguajes, lo que en este proyecto que se hace uso de varios, permite el poder distinguir y observar errores más fácilmente, realizar un código más legible, etc. Se ha trabajado con los lenguajes XHTML, CSS, Java Script, PHP y AJAX.

5.1.1 XHTML

XHTML, es el acrónimo inglés de eXtensible Hypertext Markup Language. Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML.

XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc.

5.1.2 CSS

Es un lenguaje formal para definir la presentación de un documento escrito en HTML o XHTML. La idea del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede estar adjunta en el mismo documento XHTML o en un documento de estilo, como es el caso elegido para el desarrollo de este proyecto. Los estilos se pueden definir en el fichero de estilo y a partir de etiquetas aplicarlos en el documento XHTML.

5.1.3 Java Script

Java Script es un lenguaje de programación interpretado, es decir, no requiere compilación. Se utiliza básicamente en páginas web, con una sintaxis similar a la de los lenguajes JAVA o C.

Todos los navegadores actuales interpretan el código Java Script integrado dentro de las webs. Para interactuar con la página web, se provee al lenguaje Java Script de una implementación DOM (una API, que permite acceder, añadir y modificar de forma dinámica el contenido estructurado en los documentos).

5.1.4 PHP

Es un lenguaje de programación interpretado que se utiliza para la creación de páginas dinámicas.

Generalmente se ejecuta en el servidor y permite obtener los datos almacenados en una base de datos, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otro tipo de programas.

Dentro de este proyecto, las páginas PHP se encuentran organizadas por carpetas con el nombre del módulo, cada una de estas tiene un nombre identificador, lo que permite reparar errores fácilmente.

5.1.5 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML, es un conjunto de tecnologías de desarrollo con tal de conseguir aplicaciones web interactivas.

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo término. De esta manera es posible realizar cambios sobre la misma página sin la necesidad de recargarla. Este hecho supone un aumento de la interactividad, velocidad y uso de la misma.

5.2 Organización final de los archivos

Se ha seguido el diseño de la arquitectura de los archivos con menores modificaciones:

Name	Type
 profesor	File folder
 recursos	File folder
 usuarios	File folder
 index.php	PHP File

Ilustración 30 Arquitectura de archivos - raíz

A continuación se detalla el contenido de cada carpeta.

5.2.1 Carpeta profesor

Incluye todos los archivos correspondientes al módulo del profesor. Contiene 2 carpetas: crear_codigo y grupos. Estas dos carpetas contienen archivos php.








Name	Type
 crear_codigo	File folder
 grupos	File folder
 codigos.php	PHP File
 guardar_cal.php	PHP File
 guardar_calif.php	PHP File
 index.php	PHP File
 ver_codigo.php	PHP File

Ilustración 31 Arquitectura de archivos - /profesor

5.2.2 Carpeta recursos

Aquí se han guardado los archivos que tanto el módulo de profesor y el de los alumnos utilizarán.








Name	Type
 bs	File folder
 codigos	File folder
 css	File folder
 images	File folder
 js	File folder
 syntaxhighlighter_3.0.83	File folder
 xml	File folder

Ilustración 32 Arquitectura de archivos - /recursos

5.2.2.1 bs

Aquí se guardaron todos los archivos correspondientes a bootstrap.

5.2.2.2 codigos

Todos los códigos publicados por el profesor son almacenados aquí, dichos archivos independientemente del tipo de lenguaje en el que estén escritos son guardados en .txt; la razón principal es permitir que el resaltador de síntesis trabaje mejor.

5.2.2.3 css

Almacena los archivos de estilos CSS, son usados a la par que los de la carpeta bs.

5.2.2.4 images

Se almacenan imágenes propias de los estilos.

5.2.2.5 js

Aquí se almacenan los archivos de javascript, los cuales son las librerías que permiten que las funciones programadas en dicho lenguaje puedan ejecutarse debidamente.

5.2.2.6 syntaxhighlighter_3.0.83

Repositorios del resaltador de sintaxis usado para visualizar los códigos publicados debidamente.

5.2.2.7 xml

Aquí se almacenan todos los archivos xml utilizados en el sistema. Dado que no se usaron bases de datos, la información de los usuarios, comentarios, y de etiquetas para el autocompletado al redactar un código se almacenaron aquí.

5.2.3 Carpeta usuarios

Esta carpeta contiene todos los archivos correspondientes al módulo de alumnos, sin embargo se decidió nombrarla usuario por que guarda el archivo entrar.php; este archivo permite el ingreso no solo a los alumnos sino también al profesor. La carpeta códigos contiene los archivos relacionados a la visualización del código por un alumno.









Name	Type
 codigos	File folder
 acceso.php	PHP File
 eliminar_cuenta.php	PHP File
 entrar.php	PHP File
 guardando_usu.php	PHP File
 index.php	PHP File
 registrarse.php	PHP File
 salir.php	PHP File

Ilustración 33 Arquitectura de archivos - /usuarios

5.3 Librerías externas

Para poder desarrollar el proyecto, se han utilizado librerías desarrolladas por otros programadores y que se distribuyen de manera gratuita u OpenSource. En los apartados siguientes se detalla cada una de estas librerías.

5.3.1 Syntax Highlighter de Alex Gorbachev

SyntaxHighlighter ejecuta en el navegador lo que significa que no importa qué tipo de servidor se esté usando. De hecho, SyntaxHighlighter se puede ejecutar de forma local en el ordenador sin ningún tipo de servidor web en absoluto.

Esto significa que se puede instalar SyntaxHighlighter en su página de inicio, blog, CMS, CD de documentación o cualquier otra página web. Literalmente, sólo toma un par de minutos para ponerlo en marcha.

5.3.2 jQuery

Librería considerada toda una API para la creación de páginas dinámicas, se usó ampliamente sobre todo para informar al usuario sobre el resultado de algunos procesos.

6. Pruebas

6.1 Introducción

En esta fase se realizará una evaluación del sistema para verificar que se satisfacen los requisitos especificados o para identificar las diferencias entre los resultados obtenidos y los esperados. Una prueba con éxito es aquella en la que se obtienen muchos defectos, y no todo lo contrario.

6.2 Tipos de pruebas

En grandes rasgos, podemos decir que existen dos tipos de pruebas, las pruebas de caja blanca y las de caja negra.

Las pruebas de caja blanca verifican los problemas relacionados con el código del software. Con esta técnica se buscan incoherencias en los valores que han de tomar las funciones o partes del código que no se llegan a ejecutar.

Las pruebas de caja negra son las encargadas de comprobar que el programa cumple con los requisitos funcionales y no funcionales decididos. En este tipo de pruebas se encuentran los errores de funcionalidad, de diseño de la interfaz de usuario, problemas de acceso a la base de datos y errores de rendimiento.

Otras pruebas que también se han llevado a cabo durante el desarrollo de la aplicación han sido las siguientes:

- a. Pruebas de unidad y de integración: cada función creada se ha probado de forma individual e integrada en la aplicación, des de la consulta más básica hasta la más compleja.
- b. Pruebas de bucles: Se ha verificado que los bucles tengan condición de entrada y de salida para cada nivel de anidamiento.
- c. Pruebas de usuarios finales de la aplicación: Durante el desarrollo de los módulos de la aplicación los clientes del proyecto han ido probando cada una de las secciones y avisando de fallos o posibles mejoras, han sido una parte activa del proceso de prueba.

6.3 Pruebas realizadas

Estas son algunas de las pruebas realizadas a la aplicación con el objetivo de comprobar su correcto funcionamiento.

Prueba 1	Verificación al sistema de login
Descripción	Realizar la autenticación de usuarios registrados y no registrados.
Resultado Esperado	El sistema debe permitir el acceso a todos los usuarios registrados y redireccionarlos según su perfil.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 16 Prueba No. 1

Prueba 2	Funcionamiento del menú principal del alumno
Descripción	Se verificarán todas las opciones y enlaces disponibles.
Resultado Esperado	Las opciones y enlaces funcionan correctamente, no deben mostrarse errores.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 17 Prueba No. 2

Prueba 3	Funcionamiento de la visualización de un código desde el perfil alumno
Descripción	Se corroborará la correcta visualización de un código y los comentarios en él. Además de corroborar el sistema de comentarios, el cual solo se podrá usar satisfactoriamente 1 vez.
Resultado Esperado	El código y sus comentarios se muestran adecuadamente, se puede comentar de forma correcta 1 vez por usuario.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 18 Prueba No. 3

Prueba 4	Funcionamiento del menú principal del profesor
Descripción	Se verificara el funcionamiento de todas las opciones posibles.
Resultado Esperado	Las opciones y enlaces funcionan correctamente, no deben mostrarse errores.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 19 Prueba No. 4

Prueba 5	Funcionamiento de la visualización de un código desde el perfil del profesor
Descripción	Se probaran todos los enlaces, se calificaran comentarios.
Resultado Esperado	Todas las opciones habilitadas deben de funcionar correctamente, la sección para comentarios se muestra debidamente; solo se muestran las opciones de calificar para aquellos comentarios sin calificar. El sistema de calificación funciona correctamente.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 20 Prueba No. 5

Prueba 6	Funcionamiento de la modificación de un código
Descripción	Se probaran todos los enlaces, se modificara un código.
Resultado Esperado	Las opciones y enlaces funcionan correctamente, no deben mostrarse errores. Y el código modificado se guarda correctamente.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 21 Prueba No. 6

Prueba 7	Funcionamiento de la publicación y redacción de un código
Descripción	Se probaran los enlaces, y se creara un nuevo código.
Resultado Esperado	Las opciones y enlaces funcionan correctamente, el código se publica correctamente.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 22 Prueba No. 7

Prueba 8	Funcionamiento de la gestión de grupos
Descripción	Se probaran los enlaces, se crearan grupos y se modificaran a gusto.
Resultado Esperado	Las opciones y enlaces funcionan correctamente, los grupos guardan y muestran a los integrantes correctamente.
Resultado Obtenido	Se cumple el resultado esperado.

Tabla 23 Prueba No. 8

7. Publicación del Sistema

Finalmente, tras realizar y obtener resultados satisfactorios en las pruebas el sitio ha sido alojado en un servidor público de internet. Las especificaciones y demás se describirán a continuación.

7.1 000WebHost

El sitio se ha alojado en 000WebHost, el cual es el mayor proveedor de alojamiento gratuito en la web. Provee alojamiento web libre con PHP y MySQL. netai.net dominio es utilizado por para alojar los sitios web de sus usuarios. Cada subdominio en una .netai.net * es administrado por un cliente diferente.

7.2 ingsw2016.netai.net

El sistema desarrollado puede ser accesado desde cualquier computadora con acceso a internet a través de la siguiente dirección:

<http://ingsw2016.netai.net/>

7.2.1 Vista general

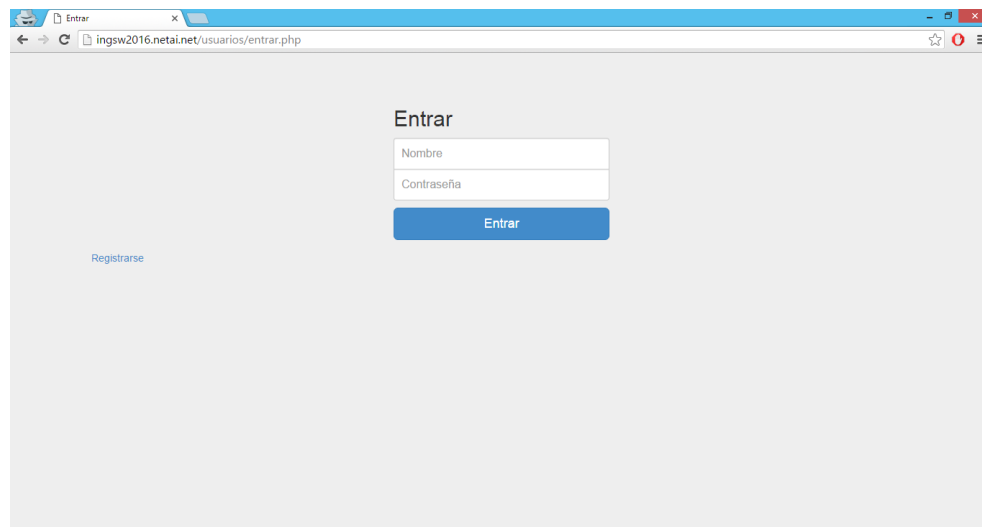


Ilustración 34 Vista general online

7. Conclusiones

8.1 Conclusiones finales

El aspecto más importante de la realización de este proyecto es el aprendizaje de las fases a seguir y la estructura necesaria para el desarrollo y finalización de un proyecto.

Estos conocimientos se podrán aplicar a cualquier aplicación que se tenga que desarrollar en un futuro.

El desarrollo de un proyecto requiere de mucho tiempo y esfuerzo; debido básicamente a que se están demostrando los conocimientos adquiridos y aplicando todos los conceptos que se aprendieron durante el curso de la asignatura.

8.2 Objetivos conseguidos

El principal objetivo del sistema es la sencilla distribución de códigos de un profesor hacia sus alumnos, permitiendo además la retroalimentación mediante comentarios de los alumnos hacia el profesor.

Además al profesor se le brindan facultades de gestión y administración al profesor lo cual le permite tener más control con respecto a las actividades docentes que realiza.

8.3 Problemas encontrados

Durante el desarrollo los problemas fueron mínimos, por una parte la gestión de la información mediante archivos XML facilitó y agilizó el proceso de diseño, pero en cuanto a la codificación se entorpeció un poco.

La sección que permite la modificación del código fue un elemento complicado de programar ya que durante el proceso de codificación constantemente se tenían problemas al volver a guardar el código.

8.4 Posibles ampliaciones

Todo sistema informático evoluciona y mejora con el tiempo, introduciendo nuevas características y funcionalidades. El sistema podría incrementar el número de profesores implicados y dotarlos con mayores facultades administrativas con respecto a que usuarios o grupos pueden acceder a cierta información.

La publicación de códigos es un aspecto clave, pese a que permite la publicación no es del todo eficaz. Un sistema que permita subir el archivo directamente a la página podría funcionar bastante bien.